

# Data Migration Facility DMF Commands

## Category: Archiving Data

At the NAS facility, certain SGI systems (primarily the Lou filesystems) support a virtual storage manager feature called DMF (Data Migration Facility). Its purpose is to allow users to keep an increased volume of data in the files under their home directories by migrating those files that are not currently in use to offline backing storage, thereby allowing active disk space to be available for active files.

Migrated files are retrieved to active disk when you attempt to read or write to them, and the whole process is substantially transparent, aside from a possible time lag while a file is being retrieved.

The process of migrating files by backing them up from disk to another medium is implemented using STK tape silos.

## DMF Commands

User commands associated with DMF are below, followed by usage examples:

dm~~l~~s      Directory listing showing file migration status.  
dm~~g~~et     Retrieve migrated files to disk.  
dm~~p~~ut     Cause files to migrate to backup on tape.  
dm~~f~~ind    Find files under a directory hierarchy.  
dm~~c~~opy    Copy all or part of offline files.  
dm~~a~~tt~~r~~    List attributes of files.

## The *dm~~l~~s* command

The *dm~~l~~s* command is much like the standard *ls* command for file or directory listing, with the addition of an option for displaying the DMF status of files. Actually, *dm~~l~~s* is derived from the GNU *ls* command, so it has a few extra "bells-and-whistles" compared to standard *ls* command; for details, see the **dm~~l~~s man page**.

Example showing the extra status field:

```
% ls -l
total 64
-rw-r----- 1 aeneuman madmag 14713 Mar 1 17:02 file1
-rw-r----- 1 aeneuman madmag 17564 Mar 1 17:02 file2
```

```
% dmls -l
total 33
-rw-r----- 1 aeneuman madmag      14713 Mar  1 17:02 (REG) file1
-rw-r----- 1 aeneuman madmag      17564 Mar  1 17:02 (REG) file2
```

DMF has several possible states for files. The first three of them are the most likely to appear:

REG	File is a "regular" file that exists only online, on active disk.
DUL	File is "dual-state": identical copies of its data exist online and offline. The online copy will persist if there is no demand for free space in its filesystem. When free space is needed, the online copy of its data is released, leaving just the offline copy; the file becomes "offline." If you make any change to a dual-state file, the offline copy becomes out of date and invalid, and the file is once again a "regular" file.
OFL	File's directory entry remains but its data blocks are located offline only.
MIG	File is in the process of migrating offline.
UNM	File is in the process of un-migrating back online.
NMG	File is nonmigratable.
INV	File's DMF state is unknown or "invalid," usually because it is in a filesystem that does not use DMF.

## The *dmget* command

The *dmget* command explicitly requests an offline file to be retrieved to disk. This command is not strictly required in order to retrieve offline files; any program that tries to use the file will cause it to be retrieved first.

The following example shows that referencing an offline file retrieves it after a pause. Notice that *C* was "offlined." Using the command *file C* caused this file to be retrieved and its status changed from *OFL* to *DUL*.

```
% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 11:20 (OFL) C

% file B
[immediate response:]
B:                English text

% file C
[pause while file is retrieved, then:]
C:                English text
```

```
% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 11:20 (DUL) C
```

You can retrieve the file explicitly with the command *dmget*. For example:

```
% dmls -l
total 220
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 11:20 (OFL) C

% dmget C ; echo Done
[pause, then:]
Done
```

One motivation for retrieving files explicitly with *dmget* is that you may be able to save time. If you are working with several files that have been migrated at the same time and reside on the same offline tape cartridge, a *dmget* command that names all the files would be able to retrieve them all in a single mount of the tape.

By contrast, simply using the files, one after another, would cause the tape robot to fetch the tape cartridge, retrieve the first file and put away the tape, then go back and fetch the cartridge, retrieve the second file, put away the tape, and so on.

Another good reason for retrieving files explicitly with *dmget* is that you control when the retrieval takes place. For example, suppose you have a large, multiprocessor application that is going to read a currently migrated data file. You would prefer the retrieval process to take place with just your shell running, not when your main application has spawned several dozen processes on several dozen nodes, with all of them having to sit idle waiting for the file to be retrieved.

See the [example under dmfind](#) for an efficient method to recall all files in a directory and its subdirectories.

## The *dmput* command

In filesystems that are under DMF automated space management, large files that have not been used for a while will be migrated offline automatically. The definitions of "large" and "a while" are established on each system by its system administrator.

You can invoke migration explicitly with the *dmput* command. Useful options are:

- r Causes DMF to release a file's online disk space immediately, giving it Offline status. Otherwise, the file would be in Dual status until its disk space was needed

for other files.

- w Causes the *dmput* command to wait until migration has completed before returning control. Otherwise the *dmput* command returns immediately with the file in Migrating status.

Here's an example showing immediate return and changing state:

```
% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 12:43 (REG) C

% dmput C ; dmls -l
[immediately:]
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 12:43 (MIG) C

[then after a while:]
% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 12:43 (DUL) C
```

Example showing delayed return, and release of disk space:

```
% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (REG) C

% dmput -r -w C ; dmls -l
[long pause:]
total 220
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
```

## The *dmfind* command

The *dmfind* command is based on the GNU version of the *find* command and adds the ability to search for files in a given DMF state. This is handy for determining which files are offline and, therefore, candidates for retrieval with *dmget*.

Example:

```
% dmls -l
total 220
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmfind . -state ofl -print
./C

% dmget `dmfind . -state ofl -print`
[pause:]

% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (DUL) C
```

To efficiently recall all files in a directory named mydir and its subdirectories, use the following command:

```
% dmfind mydir -state mig -or -state ofl -print | dmget
```

## The *dmcopy* command

The *dmcopy* command copies all or part of an offline file to a destination file, keeping the offline file offline.

When the offline file is being copied in its entirety, the only arguments needed are the two filenames.

Example:

```
% dmls -l
total 224
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmcopy C newC
[pause:]

% dmls -l
total 1404
-rw-r----- 1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r----- 1 aeneuman madmag      201550 Mar  2 11:24 (REG) B
-rw-r----- 1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
-rw----- 1 aeneuman madmag    1209300 Mar  4 15:06 (REG) newC
```

*Dmcopy* has options that allow copying just part of the offline file, and specifying offset locations in the source and destination files.

-l length	Specifies a data length to copy in bytes. The default is the whole size of the source file.
-o source-offset	Specifies a byte offset in the offline source file where reading is to begin. The default is zero; that is, reading begins at the start of the source file.
-d destination-offset	Specifies a byte offset in the destination file where writing is to begin. Any bytes in the destination file before this offset are zero filled. The default destination offset is whatever the source offset is.

Example: Skip two records of the offline source file and copy the next three records. Records are 2K bytes long.

```
% set RECDLEN=2048
% set NRECD=3
% @ LENGTH = $NRECD * $RECDLEN
% set NSKIP=2
% @ OFFSET = $NSKIP * $RECDLEN

% dmcop -l $LENGTH -o $OFFSET -d 0 C newC
[pause:]

% dmls -l *C
-rw-r----- 1 aeneuman madmag      1209300 Mar  3 15:17 (OFL) C
-rw----- 1 aeneuman madmag         6144 Mar  4 16:10 (REG) newC
```

**Note:** If the source file is a regular file, without an offline copy in DMF, the destination file is always zero length.

**Note:** The modes on the source file are ignored when creating the destination file. The modes on the destination file are 0666 (readable and writable by everyone), except where blocked by the current umask. Review your output files' permissions and reset them with *chmod* as needed.

## The *dmattr* command

The *dmattr* command prints selected attributes of specified files. This is most useful in shell scripts.

Some options include:

-a attr1,attr2,...	Selects a subset of the reportable attributes.
-d delimiter	Specifies a one-character separator between adjacent values. A space is the default.
-l	Prints the attributes, one per line, with labels.

## Examples:

```
% dmattr -l A
    bfid : 0
    emask : 0
    fhandle : 010000000000000188dede3a4b319c5b9000e000000000001000000000b3fd163
    flags : 0
    nreg : 0
    owner : 4771
    path : A
    size : 20155
    space : 20480
    state : REG

% dmattr A
0 0 010000000000000188dede3a4b319c5b9000e000000000001000000000b3fd163 0 0
4771 A 20155 20480 REG

% dmattr -a owner,state -d : A
4771:REG

% foreach file ( * )
? if (`dmattr -a state $file` == OFL) then
? echo $file is offline
? endif
? end

C is offline
```

---

Article ID: 250

Last updated: 25 May, 2011

Data Storage & Transfer -> Archiving Data -> Data Migration Facility DMF Commands

<http://www.nas.nasa.gov/hecc/support/kb/entry/250/?ajax=1>